

This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

**As rescanning documents *will not* correct images,
please do not report the images to the
Image Problem Mailbox.**



PATENT ABSTRACTS OF JAPAN

(11) Publication number: **05216684 A**(43) Date of publication of application: **27 . 08 . 93**(51) Int. Cl. **G06F 9/45**(21) Application number: **04021056**(71) Applicant: **FUJITSU LTD**(22) Date of filing: **06 . 02 . 92**(72) Inventor: **SUGIMOTO HIROYUKI**(54) **PROGRAM EXECUTION DEVICE AND EXECUTION METHOD**

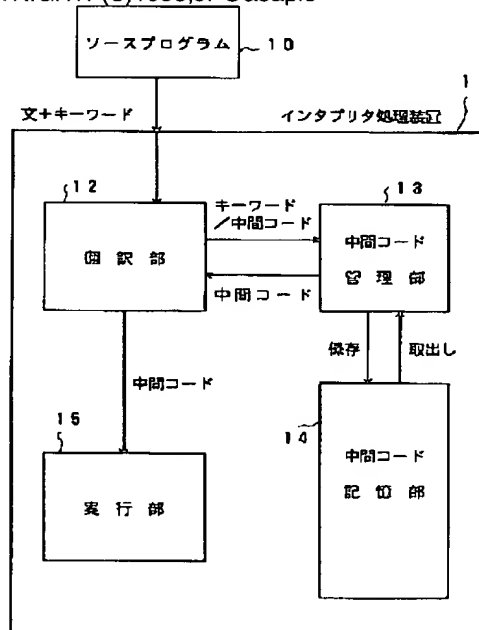
(57) Abstract:

PURPOSE: To improve the execution performance of an interpreter by decreasing translation time by reutilizing the code of an intermediate language translated in the interpreter.

CONSTITUTION: A translation part 12 inputs a sentence or a sentence string which is a translation unit in a source program 10, and inputs or prepares a keyword discriminating it uniquely in the source program 10. A code converted into an intermediate language is retrieved by using an intermediate code and whether the code is preserved in an intermediate code storage part 14 or not is investigated. As a result of the retrieval, when the code converted into the intermediate language exists, the code of the intermediate language is taken out from an intermediate code storage part 14 by a keyword control part 13 by the processing of an intermediate code control part 13. Then, an execution part 15 translates and executes the code converted into the intermediate language. Namely, when the same keyword as the one preserved in the intermediate code storage part 14 appears, the keyword of the intermediate

language is reutilized.

COPYRIGHT: (C)1993,JPO&Japio



(19) 日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平5-216684

(43) 公開日 平成5年(1993)8月27日

(51) Int. Cl. ⁵

G 0 6 F 9/45

識別記号

庁内整理番号

F I

技術表示箇所

9292-5 B

G 0 6 F 9/44 3 2 2 E

9292-5 B

3 2 0 C

審査請求 未請求 請求項の数 2

(全 8 頁)

(21) 出願番号 特願平4-21056

(22) 出願日 平成4年(1992)2月6日

(71) 出願人 000005223

富士通株式会社

神奈川県川崎市中原区上小田中1015番地

(72) 発明者 杉本 裕之

神奈川県川崎市中原区上小田中1015番地

富士通株式会社内

(74) 代理人 弁理士 小笠原 吉義 (外2名)

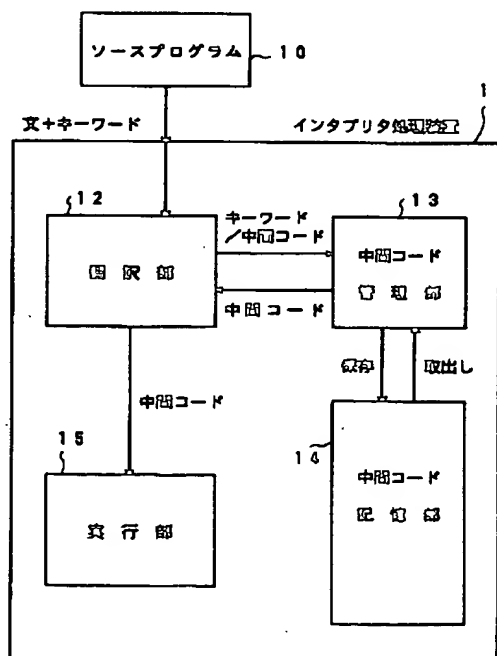
(54) 【発明の名称】 プログラム実行装置および実行方法

(57) 【要約】

【目的】 ソースプログラム10を中間言語のコードに変換して、それを解釈し実行するインタプリタ方式によるプログラム実行装置および実行方法に関し、翻訳した中間コードを再利用することにより翻訳時間を減らし、インタプリタの実行性能を向上させることを目的とする。

【構成】 ソースプログラム10中の文または文の列を一意的に識別するためのキーワードと中間言語に変換したコードとを対応させて管理する中間コード管理部13を設け、翻訳部12は、キーワードを使用することにより、一度実行した文または文の列の中間言語に変換したコードを再利用する。

本発明の凸本構成図



【特許請求の範囲】

【請求項1】 ソースプログラム(10)を中間言語のコードに変換して、それを解釈し実行するインタプリタ方式によるプログラム実行装置において、ソースプログラム中の文または文の列を中間言語に変換したコードを、その文または文の列をソースプログラム中で一意的に識別するためのキーワードと共に記憶する中間コード記憶部(14)と、前記キーワードを使用し、前記中間コード記憶部への中間言語に変換したコードの保存および取り出しを行う中間コード管理部(13)と、ソースプログラムの文または文の列を入力し、その文または文の列に関するキーワードに対応する中間言語に変換したコードが既に前記中間コード記憶部に記憶されていれば、それを前記中間コード管理部によって取り出し、前記中間コード記憶部に記憶されていなければ、入力した文または文の列を中間言語のコードに変換して、その変換結果を前記中間コード管理部によって前記中間コード記憶部に保存する翻訳部(12)と、前記翻訳部から通知された中間言語のコードを解釈して実行する実行部(15)とを備えたことを特徴とするプログラム実行装置。

【請求項2】 ソースプログラム(10)を中間言語のコードに変換して、それを解釈し実行するインタプリタ方式によるプログラム実行方法において、ソースプログラム中の翻訳単位である文または文の列を入力し、かつそれをソースプログラム中で一意的に識別するキーワードを入力または作成する処理過程と、前記キーワードによって、中間言語に変換したコードを検索し、所定の記憶域に保存されているか否かを調べる処理過程と、検索した結果、中間言語に変換したコードがない場合に、入力した翻訳単位の文または文の列を中間言語のコードに変換し、その変換結果をキーワードと共に所定の記憶域に保存する処理過程と、検索した結果、中間言語に変換したコードがある場合に、それを所定の記憶域から取り出す処理過程と、中間言語に変換したコードを解釈し実行する処理過程とを備え、ソースプログラムにおける既に実行した文または文の列に関する中間言語のコードを再利用することを特徴とするプログラム実行方法。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は、ソースプログラムを中間言語のコードに変換して、それを解釈し実行するインタプリタ方式によるプログラム実行装置および実行方法に関する。

【0002】高水準プログラミング言語で記述されたプログラムの実行形態として、そのソースプログラムを事前に実行形式のコードに変換しておき、それを実行するコンパイラ方式と、入力したソースプログラムを翻訳し直ちに実行するインタプリタ方式とがある。一般にインタプリタ方式による実行は、コンパイラ方式による場合に比べて処理時間が長くなるので、その高速化が望ま

れている。

【0003】

【従来の技術】高水準プログラミング言語を計算機で実行する場合、計算機プログラムを実行時に翻訳して実行するインタプリタは、コンパイラによって実行形式のコードに変換し、それを計算機で直接実行する方式に比べて、柔軟で使いやすいという特長を持っている。しかし、実行時に翻訳を行うために、インタプリタの実行性能は、コンパイラによる方式に比べてかなり悪いのが一般的である。

【0004】

【発明が解決しようとする課題】本発明は上記問題点の解決を図り、インタプリタにおいて翻訳した中間言語のコード(以下、中間コードと呼ぶ)を再利用することにより、翻訳時間を減らし、インタプリタの実行性能を上げることを目的としている。

【0005】

【課題を解決するための手段】図1は本発明の基本構成図である。インタプリタ処理装置11は、CPUおよびメモリなどからなる装置であり、高水準プログラミング言語で記述されたソースプログラム10を翻訳し実行する装置である。本発明に係るインタプリタ処理装置11は、翻訳部12、実行部15、中間コード管理部13および中間コード記憶部14を備えている。

【0006】翻訳部12は、翻訳・実行単位となるソースプログラム10中の文または文の列と、それをソースプログラム10で一意的に識別できるようなキーワードを読み込み、文または文の列を中間コードに変換する。中間コードとは、機械語命令列によるコードまたは実行しやすい形に変換された所定の内部規約によるコードである。実行部15は、その中間コードを解釈して実行する処理手段である。

【0007】中間コード記憶部14は、中間コードをキーワードと共に記憶しておく部分である。中間コード管理部13は、キーワードを使って、中間コード記憶部14内にある中間コードを検索したり、中間コード記憶部14内に中間コードとキーワードとを保存したりする処理手段である。

【0008】翻訳部12は、ソースプログラム10中の翻訳単位である文または文の列を入力すると共に、それをソースプログラム10中で一意的に識別するキーワードを入力または作成する。そして、中間コード管理部13による処理により、キーワードを用いて、中間言語に変換したコードを検索し、中間コード記憶部14に保存されているか否かを調べる。

【0009】検索した結果、中間言語に変換したコードがない場合に、翻訳部12は、入力した翻訳単位の文または文の列を中間言語のコードに変換し、その変換結果をキーワードと共に、中間コード管理部13の処理により保存する。

【0010】検索した結果、中間言語に変換したコードがある場合には、中間コード管理部13により、中間言語のコードを中間コード記憶部14から取り出す。実行部15は、中間言語に変換したコードを解釈し実行する。

【0011】以上のように、ソースプログラム10における既に実行した文または文の列に関する中間言語のコードを、中間コード記憶部14に保存し、同じキーワードが出現した場合に、その中間言語のコードを再利用する。

【0012】

【作用】インタプリタ処理装置11は、入力したソースプログラム10中の文または文の列を翻訳した中間コードを、キーワードにより管理する機構を持つ。一度実行した文に対応する中間コードを、キーワードと共に中間コード記憶部14に保存し、同じ文を再び実行する場合には、保存しておいた中間コードを取り出して実行する。したがって、一度実行した文に対する翻訳処理が不要になり、繰り返し文の中の文のように、何度も実行される文を高速に実行することができるようになる。

【0013】

【実施例】図2は本発明の実施例フローチャートを示す。以下、図2に示す処理□～□に従って説明する。

【0014】□ まず、翻訳部12が、次の実行単位となる文と、ソースプログラム10中で一意的に識別するためのキーワードを入力する。なお、キーワードをソースプログラム10とは別に入力するのではなく、ソースプログラム10から抽出するなどの処理により、翻訳部12が自動生成するようにしてもよい。

【0015】□ 翻訳部12は、中間コード記憶部14に、キーワードに対応する中間コードが存在するかどうかを、中間コード管理部13に問い合わせる。

□ 中間コードが中間コード記憶部14に存在すれば、処理□へ進み、存在しなければ、処理□へ進む。

【0016】□ 翻訳部12は、入力した文の翻訳を行い、中間コードを生成する。

□ 翻訳結果の中間コードとキーワードとを中間コード管理部13に送り、それらを中間コード記憶部14に格納して記憶する。その後、処理□へ進む。

【0017】□ 中間コード記憶部14に該当する中間コードが存在する場合、翻訳部12は、中間コード管理部13の処理により、キーワードに対応する中間コードを取り出す。

【0018】□ 翻訳部12から実行部15へ中間コードを送り、実行部15は中間コードを解釈実行する。実行部15の処理は、従来と同様である。図3は本発明の実施例を説明するためのソースプログラムの例を示す図、図4は図3に示すソースプログラムが扱うテーブル構造の例を示す図である。また、図5は本発明の実施例の構成図である。

【0019】図3の(イ)はソースプログラムであり、(ロ)はそれに付与されたキーワードの例である。キーワードは、対応するソースコードと同じ行番号位置

(1:~6:)に書かれている。この例では、キーワードは、複数のキーの組になっている。最初の数字はキーの数で、それ以降に複数のキーがコンマで区切って並べられる。このソースプログラムは、図4の(ロ)に示すテーブルTBL2のカラムの内容に従って、テーブルTBL1のカラムを更新するプログラムである。

10 【0020】図5において、図1と同符号のものは図1に示すものに対応し、50はインタプリタ制御部、51はソースコードを中間コードに変換する変換部、52はキーワードによるハッシュ表、53は中間コードを記憶する中間コード記憶域を表す。

【0021】B1およびB2は、インタプリタに入力されたソースプログラムのコードおよびそのキーワードを保持するためのバッファである。また、B3は中間コードを一時的に保持するバッファである。インタプリタ制御部50は、インタプリタ全体の動作を制御する。変換部51は、インタプリタ制御部50の指示により、入力されたソースコードをバッファB1から読み込んで中間コードに変換し、バッファB3に格納する。

20 【0022】実行部15は、インタプリタ制御部50の指示により、バッファB3の内容を取り出して実行する部分である。また、中間コード管理部13は、インタプリタ制御部50の指示に従い、中間コード記憶部14内の中間コードの検索や中間コードの保存を行う。中間コード記憶部14は、保存している中間コードをキーワードによって高速に検索できるように、ハッシュ表52と中間コード記憶域53の2つに分かれている。

30 【0023】次に、図3に示すソースプログラムの例に従って、図5に示すインタプリタの動作を説明する。図3の(イ)に示すソースプログラムは、データベース検索言語としてよく知られているSQL言語風に記述されたもので、1~6の各行は、以下のような処理内容になっている。

【0024】

1: テーブルTBL1の内容を表示する。

2: テーブルTBL2の内容を表示する。

3: テーブルTBL2のカラムXのすべての値(Xの値をxとする)、について次の文を実行する。

【0025】4~5: テーブルTBL1において、カラムAの値がカラムBとカラムCの値の和より小さいような行に対して、カラムxの値を1だけ小さくする。

6: テーブルTBL1の内容を表示する。

40 【0026】(i) まず、インタプリタは、1:の文「SELECT * FROM TBL1;」と、それに対応するキーワード「2, SELECT, TBL1」を入力して、それぞれバッファB1とバッファB2に格納する。

【0027】(2) 中間コード管理部13は、バッファB2の内容を読み込んで、中間コード記憶部14の中に、バッファB2の内容をキーに持つ中間コードがあるかどうかを検索する。

【0028】(3) 最初は、中間コードが記憶されていないので、検索に失敗し、インタプリタ制御部50に検索が失敗したことが通知される。インタプリタ制御部50は、変換部51にバッファB1の内容を中間コードに変換することを指示する。

【0029】(4) 変換部51は、バッファB1の内容を読み込んで、それを中間コードに変換し、バッファB3に格納する。

(5) 上記処理(3)で検索が失敗しているので、インタプリタ制御部50は、中間コード管理部13に、バッファB2とB3の内容を中間コード記憶部14に保存するよう指示する。

【0030】(6) 中間コード管理部13は、バッファB2とB3の内容を読み込んで、バッファB3の内容を中間コード記憶部53に保存し、また、バッファB2の内容をキーとして、対応する中間コードが検索できるようにハッシュ表52を更新する。

【0031】(7) 次に、インタプリタ制御部50は、実行部15にバッファB3の内容を実行するように指示する。

(8) 実行部15は、バッファB3の内容を読み込み、実行する。

【0032】以上で、1:の文の実行が完了する。2:の文も同様である。次に、6:の文のように、既に同じキーワードを持つ文(この例では1:の文)が実行されているような場合について説明する。

【0033】(1)~(2)は、上記1:の文の処理と同様である。

(3) バッファB2の内容と同じキーワードを持つ中間コードの検索に成功するので、検索に成功したことがインタプリタ制御部50に通知される。

【0034】(4) インタプリタ制御部50は、中間コード管理部13に、バッファB2のキーワードに対応する中間コードを、バッファB3に格納するように指示する。

(5) 中間コード管理部13は、バッファB2に対応した中間コードをバッファB3に格納する。

【0035】(6) インタプリタ制御部50は、実行部15に対し、バッファB3の内容を実行するように指示する。

(7) 実行部15は、バッファB3の内容を実行する。

【0036】以上のように、同じキーワードを持つ文は、中間コードへの翻訳処理を省略して、実行すること

ができる。また、この実施例では、図3に示す4:~5:のUPDATE文のように、キーワード自体に変数xを指定することもできる。この場合には、バッファB2に読み込まれたキーワードを使用して、中間コード管理部13が中間コードを検索する際に、キーワード中のxを現在のxの値で置き換えて、キーワード検索を行う。このようにすることにより、実行時でないと実行文が決まらないような動的なプログラムの実行でも、中間コードの再利用が可能になる。

10 【0037】ソースプログラムに対するキーワードの指定は、必ずしもすべての文に対して行う必要はなく、翻訳に時間がかかるような特定の文または繰り返し実行されるような文に対してのみ行うようにしてもよい。

【0038】例えば、データベースの検索要求があった場合に、その検索プログラムを、他の言語で記述されたプログラムにより動的に作成して、そのインタプリタを呼び出すことにより実行する、いわゆる動的SQLと呼ばれるような言語処理系においては、あらかじめ検索内容が決まっている静的SQLとは違い、コンパイラによる実行方式を採ることはできない。したがって、この

20 ような場合に本発明を適用すると、検索処理の高速化に効果がある。この例のような場合、キーワードをプログラム作成時に自動付与することも可能である。

【0039】

【発明の効果】以上説明したように、本発明によれば、中間言語方式のインタプリタでプログラムを実行する場合に、繰り返し実行する文または文の列の翻訳が一度で済むため、インタプリタの実行性能が向上する。また、本発明では、実行前に中間コードに翻訳して中間コード

30 を再利用する方式に比べ、実行時に翻訳することが可能であるため、より柔軟な中間コードの再利用が可能になる。

【図面の簡単な説明】

【図1】本発明の基本構成図である。

【図2】本発明の実施例フローチャートである。

【図3】本発明の実施例を説明するためのソースプログラムの例を示す図である。

【図4】図3に示すソースプログラムが扱うテーブル構造の例を示す図である。

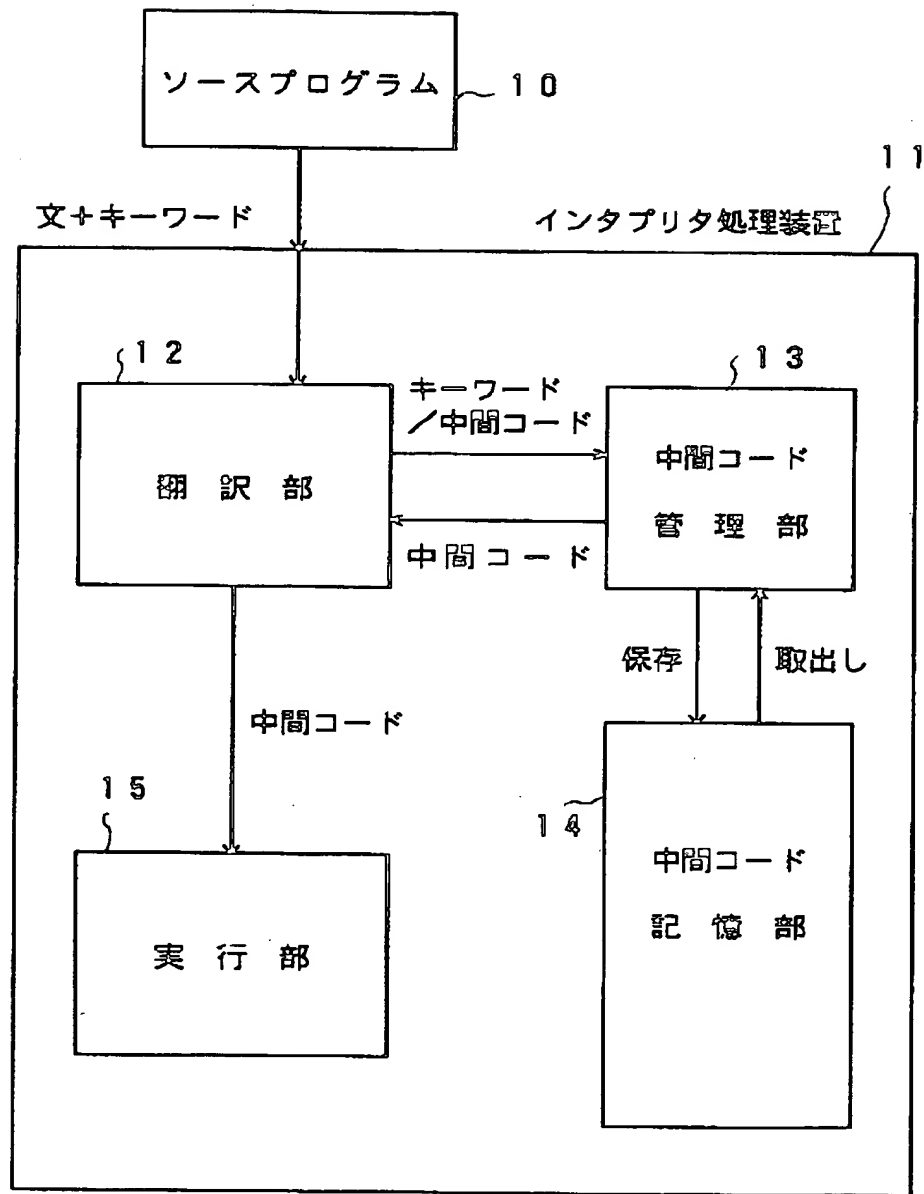
40 【図5】本発明の実施例の構成図である。

【符号の説明】

- 10 ソースプログラム
- 11 インタプリタ処理装置
- 12 翻訳部
- 13 中間コード管理部
- 14 中間コード記憶部
- 15 実行部

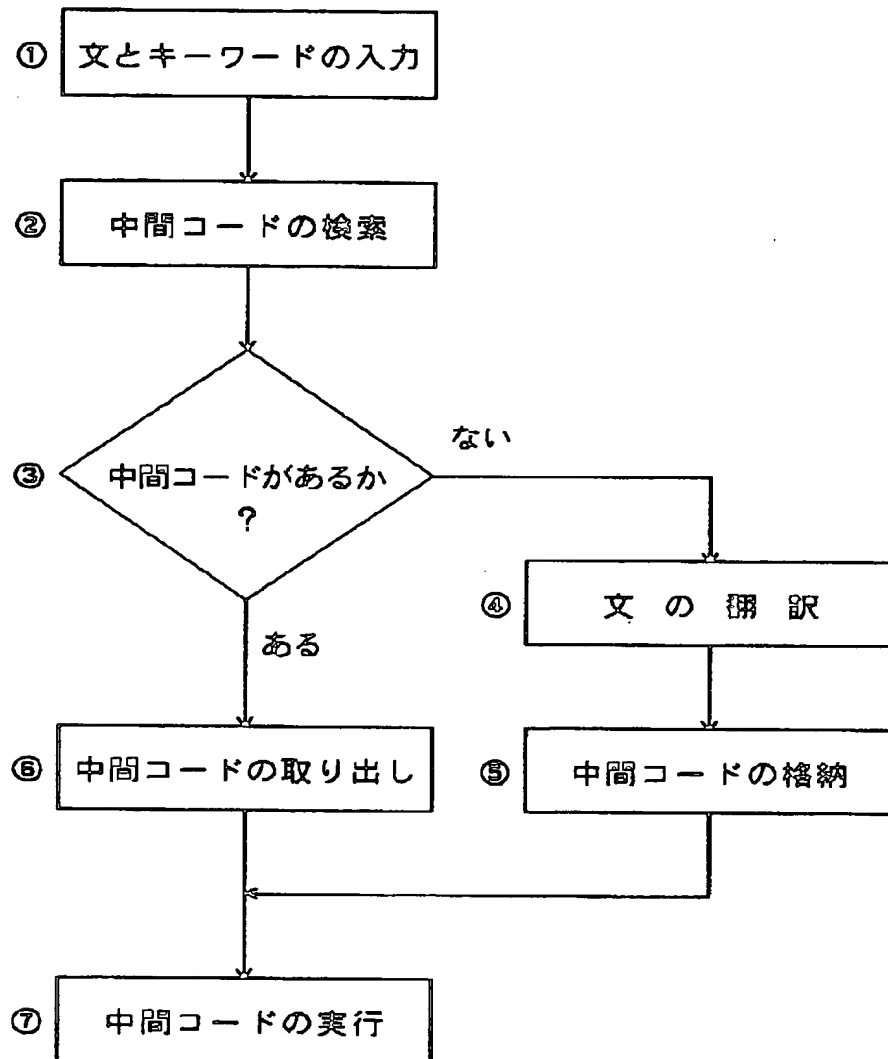
【図1】

本発明の基本構成図



【図2】

実施例フローチャート



【図3】

ソースプログラムの例

(イ) ソースプログラム

```

1: SELECT * FROM TBL1;
2: SELECT * FROM TBL2;
3: FOR ALL X:x FROM TBL2
4:   UPDATE TBL1 SET x=x-1
5:   WHERE A<B+C;
6: SELECT * FROM TBL1

```

(ロ) キーワード

```

1: 2, SELECT, TBL1
2: 2, SELECT, TBL1
3:
4: 2, UPDATE, x
5:
6: 2, SELECT, TBL1

```

【図4】

テーブル構造の例

(イ) TBL1

| A | B | C |
|---|---|---|
| 5 | 5 | 3 |
| 2 | 0 | 0 |
| 0 | 3 | 2 |
| 3 | 0 | 1 |

(ロ) TBL2

| X |
|---|
| B |
| C |
| B |

【図 5】

実施例の構成図

